Lab assignment
CPE 357
Systems Programming

CAL POLY
SAN LUIS OBISPO

Lab 2

Exercise:

Write a program to change the contrast of a bitmap image.
The image files should be 24Bit per pixel (standard) BMP files. You will find several ones in the adjunkt zip to this assigment. You can use your own ones – save them as 24Bit BMP in e.g. photoshop.

For looping through the pixels and changing the contrast, use a function!

The Program:

The program should read several parameters from the comand line:
[programname] [imagefile1] [outputfile] [contrastfactor]
e.g.

```
./contrast face.bmp resultface.bmp 3.2
```

Catching wrong or missing parameters not necessary.

To allocate dynamically the necessary memory for storing the pixel data, use brk() and/or sbrk().

Contrast:
Pixel (normalized) = pow (Pixel (normalized), factor);

Howto:

First read the bitmap file header infos into corresponding structures and then the image data into an array of BYTES (unsigned char). You need to allocate this array first.
The size is dependend on the resolution, which you get with the file headers:
http://www.dragonwins.com/domains/GetTechEd/bmp/bmpfileformat.htm
Wikipedia isn't bad here, but a bit chaotic:
https://en.wikipedia.org/wiki/BMP_file_format

Important!
Color tables and bit masks are not necessary, that's too complex. So all optional BMP data will be skipped!

Structures for BMP Format

```
typedef unsigned short WORD;
typedef unsigned int DWORD;
typedef unsigned int LONG;
struct tagBITMAPFILEHEADER
       {
       WORD bfType;  //specifies the file type
       DWORD bfSize;  //specifies the size in bytes of the bitmap file
       WORD bfReserved1;  //reserved; must be 0
       WORD bfReserved2;  //reserved; must be 0
       DWORD bfOffBits;  //species the offset in bytes from the bitmapfileheader to
the bitmap bits
       };
struct tagBITMAPINFOHEADER
       {
       DWORD biSize;  //specifies the number of bytes required by the struct
       LONG biWidth;  //specifies width in pixels
       LONG biHeight;  //species height in pixels
       WORD biPlanes; //specifies the number of color planes, must be 1
       WORD biBitCount; //specifies the number of bit per pixel
       DWORD biCompression;//spcifies the type of compression
       DWORD biSizeImage;  //size of image in bytes
       LONG biXPelsPerMeter;  //number of pixels per meter in x axis
       LONG biYPelsPerMeter;  //number of pixels per meter in y axis
       DWORD biClrUsed;  //number of colors used by th ebitmap
       DWORD biClrImportant;  //number of colors that are important
       };
```

Possible Hazzards:

Padding issues with reading the structures and padding issues per line of the images!